IN THE CLAIMS

1. (Currently amended) A translator for translating a source file in a source format to a target file in a target format, the translator comprising:

a feature identifier to determine a feature set of the source file; ~~and~~

a buffer to assemble the feature set; and

a feature writer to write the feature set into the target file in the target format.

2 (Cancelled)

3. (Cancelled)

4. (Currently amended) The translator of claim 1, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, ~~implied~~ user-created features, implied feature endings, and combinations thereof.

5. (Original) The translator of claim 1, wherein the feature identifier comprises a front-end converter to map code fragments of the source file to a list of features.

6. (Original) The translator of claim 5, wherein the front-end converter comprises a front-end lookup table.

7. (Original) The translator of claim 6, wherein the front-end lookup table is user modifiable.

8. (Original) The translator of claim 1, wherein the feature writer comprises a back-end converter to map the feature set to code fragments of the target file format.

9. (Original) The translator of claim 8, wherein the back-end converter comprises a back-end lookup table.

10. (Original) The translator of claim 5, comprising a plurality of feature writers to write the feature set into a plurality of target files having a plurality of target formats.

11. (Original) The translator of claim 1, comprising a plurality of feature identifiers to determine a feature set of a plurality of source files having a plurality of source formats.

12. (Original) The translator of claim 5, wherein the front-end converter comprises a lexical analyzer to identify tokens disposed within the source file, and a feature collector to associate the tokens with the feature set.

13. (Original) The translator of claim 1, further comprising a user interface.

14. (Original) The translator of claim 13, wherein the user interface comprises a GUI.

15. (Original) The translator of claim 1, further comprising a source format adapter module to interface with a source file generator.

16. (Original) The translator of claim 15, wherein the source format adapter module enables the source file generator to initiate translation by the translator.

17. (Original) The translator of claim 1, further comprising a target file adapter module to perform secondary translation.

18. (Original) The translator of claim 17, wherein the target file adapter module translates the target file into another target format.

19. (Original) The translator of claim 1, wherein the source and target formats are selected from the group consisting of MIF, RTF, WordPerfect, VENTURA, Microsoft Word, Interleaf, HTML, SGML, XML, C, C++, Visual Basic, Pascal, Java, MFC, PowerPlant, Swing, SVG, HPJ, Flash, WMF, VRML, RenderMan, 3DMF, and combinations thereof.

20. (Currently Amended) A method of translating a file from a source format to a target format, the method comprising:

(a) identifying a feature set of a source file; ~~and~~

(b) assembling the feature set in a buffer; and

(c) writing the feature set into a target file in the target format.

21. (Cancelled)

22. (Currently amended) The method of claim 20, wherein features of the feature set include at least one of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, ~~implied~~ user-created features, implied feature endings, and combinations thereof.

23. (Original) The method of claim 20, wherein the identifying step (a) comprises mapping code fragments of the source file to a feature list.

24. (Original) The method of claim 23, wherein the identifying step (a) comprises

looking up the code fragments in a front-end lookup table.

25. (Original) The method of claim 24, further comprising permitting the front-end lookup table to be user modifiable.

26. (Original) The method of claim 20, wherein the writing step (b) comprises mapping the feature set to code fragments of the target file format.

27. (Original) The method of claim 26, wherein the writing step (b) comprises looking up the feature set in a back-end lookup table.

28. (Original) The method of claim 20, wherein the writing step (b) comprises writing the feature set into a plurality of target files having a plurality of target formats.

29. (Original) The method of claim 20, wherein the identifying step (a) comprises identifying a feature set of a plurality of source files having a plurality of source formats.

30. (Original) The method of claim 20, wherein the identifying step (a) comprises identifying tokens disposed within the source file, and associating the tokens with the feature list.

31. (Original) The method of claim 20, further comprising using a source file generator to initiate translation by the translator.

32. (Original) The method of claim 20, further comprising using a target file adapter module to perform secondary translation.

33. (Original) The method of claim 32, wherein the target file adapter module translates

the target file into another target format.

34. (Currently amended) A method of configuring a system to translate a source file in a source format to a target file in a target format, the method comprising:

(a) providing a feature identifier to determine a feature set of the source file; ~~and~~

(b) providing a buffer to assemble the feature set; and

(c) providing a feature writer to write the feature set into the target file in the target format.

35. (Currently amended) A system for translating a source file in a source format to a target file in a target format, the system comprising:

a feature identifier to determine a feature set of the source file; ~~and~~

a buffer to assemble the feature set; and

a feature writer to write the feature set into the target file in the target format.

36. (Currently amended) An article of manufacture for translating a source file in a source format to a target file in a target format, the article of manufacture comprising:

a computer usable medium having a computer readable program code embodied therein, the computer usable medium having:

computer readable program code for identifying a feature set of the source file; ~~and~~

computer readable program code for assembling the feature set; and

computer readable program code for writing the feature set into the target file in the target format.

37. (Currently amended) Computer readable program code for translating a source file in a source format to a target file in a target format, the computer readable program

code comprising:

computer readable program code for identifying a feature set of the source file; ~~and~~

computer readable program code for assembling the feature set; and

computer readable program code for writing the feature set into the target file in the target format.

38. (Original) A translator for translating a source file in an MIF format to a target file in an HTML format, the translator comprising:

a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file;

a buffer to store the feature set; and

a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format.